
บริษัท สำนักพิมพ์เอมพันธ์ จำกัด
AIMPHAN PRESS CO., LTD.

หนังสือคู่มือการทดลอง

ชุดฝึกการเรียนรู้

การเขียนโปรแกรมอาดุยโน่เบื้องต้น

Start fun for Arduino from AimPhan Equipment

ผู้แต่ง

ผู้ช่วยศาสตราจารย์ กีรติษ สายพัทลุง วศบ. วิศวกรรมไฟฟ้า (ป.ตรี)
วศม. วิศวกรรมไฟฟ้า (ป.โท)

บรรณาธิการ

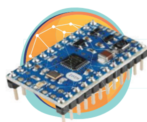
สุชาดา วราหพันธ์ กศ.บ. (สังคมศึกษา), ค.ม. (พื้นฐานการศึกษา)


จัดพิมพ์โดย บริษัท สำนักพิมพ์เอมพันธ์ จำกัด

ISBN: 000 000 00 000 0

ฝ่ายการตลาด, ฝ่ายผลิตและจัดส่ง, ฝ่ายการเงินและบัญชี :
69/109 หมู่ 1 ซ.พระแม่การุณย์ ต.บ้านใหม่ อ.ปากเกร็ด จ.นนทบุรี 11120
โทร. 0 2584 5889, 0 2584 5993, 0 2961 4580-2, โทรสาร. 0 2961 5573, 0 2582 2313

ฝ่ายวิชาการ :
87/122 ถ.เทศบาลสงเคราะห์ แขวงลาดยาว เขตจตุจักร กรุงเทพฯ 10900
โทร. 0 29544818-20, 0 29538168-9 โทรสาร. 0 2580 2923



คำนำ

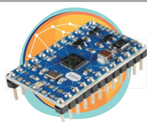
บริษัท สำนักพิมพ์เอมพันธ์ จำกัด ได้จัดทำชุดการเรียนรู้เริ่มต้นการเขียนโปรแกรมและการต่อวงจรสำหรับอาดุยโน (Arduino) ขึ้น เพื่อพัฒนาศักยภาพการเรียนการสอนวิชาที่เกี่ยวข้องกับการเขียนโปรแกรมคอมพิวเตอร์ วิทยาการหุ่นยนต์ ไมโครคอนโทรลเลอร์ ที่เหมาะสมกับการนำไปใช้งานในระดับประกาศนียบัตรวิชาชีพ ระดับประกาศนียบัตรวิชาชีพชั้นสูง ระดับปริญญาตรี และนักเรียนในระดับมัธยมศึกษาทั้งตอนต้น และตอนปลาย โดยเนื้อหาในหนังสือจะมีทั้งสิ้น 8 บท ประกอบด้วยเรื่อง

- บทที่ 1 ความรู้เบื้องต้นเกี่ยวกับอาดุยโน (Arduino) และการเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์
- บทที่ 2 ซอฟต์แวร์ภาษาซีสำหรับพัฒนาสำหรับโปรแกรมอาดุยโน
- บทที่ 3 การเขียนโปรแกรมสำหรับอุปกรณ์แสดงผล
- บทที่ 4 การเขียนโปรแกรมรับค่าจากเซนเซอร์ผ่านพอร์ตดิจิทัล
- บทที่ 5 การเขียนโปรแกรมรับค่าจากเซนเซอร์ผ่านพอร์ตแอนาล็อก
- บทที่ 6 การเขียนโปรแกรมควบคุมมอเตอร์
- บทที่ 7 การประยุกต์ใช้งานเซนเซอร์ในลักษณะต่าง ๆ
- บทที่ 8 การประยุกต์ใช้งานในการควบคุม Smart Home

ซึ่งผู้เขียนหวังว่าหนังสือ และสื่อการเรียนการสอนเล่มนี้จะเป็นประโยชน์ต่อผู้เรียนและอาจารย์ผู้สอน นำไปใช้เป็นแนวทางในการจัดการเรียนการสอนตามหลักการ เนื้อหา และการพัฒนาการเรียนรู้ให้ทันสมัยอยู่เสมอ โดยเน้นกระบวนการคิด วิเคราะห์ ฝึกแก้ปัญหา และการลงมือปฏิบัติจริง หากมีข้อเสนอแนะประการใด บริษัท สำนักพิมพ์เอมพันธ์ จำกัด ยินดีน้อมรับไว้ด้วยความขอบคุณยิ่ง



บริษัท สำนักพิมพ์เอมพันธ์ จำกัด

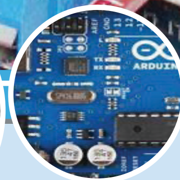


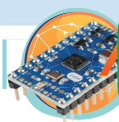
สารบัญ

บทที่ 1

ความรู้เบื้องต้นเกี่ยวกับอาดุยโน (Arduino) และการเขียนโปรแกรมภาษาซี สำหรับไมโครคอนโทรลเลอร์

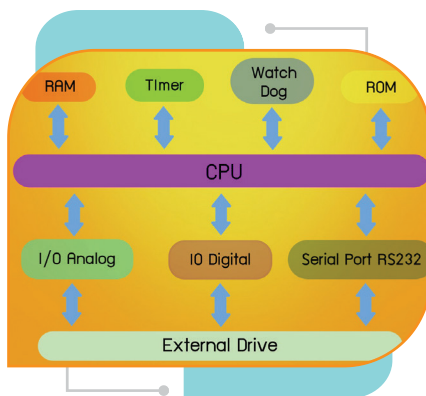






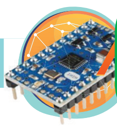
1 ความรู้เกี่ยวกับไมโครคอนโทรลเลอร์เบื้องต้น

ไมโครคอนโทรลเลอร์ (อังกฤษ: Micro-controller มักย่อว่า µC, uC หรือ MCU) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งลักษณะใกล้เคียงกับระบบคอมพิวเตอร์ ซึ่งไมโครคอนโทรลเลอร์ได้รวมซีพียู หน่วยความจำทั้งหน่วยความจำถาวร และหน่วยความจำชั่วคราว พอร์ตดิจิทัลอินพุต/เอาต์พุต (Input/Output) พอร์ตอนาล็อกอินพุต/เอาต์พุต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน ไว้ตัวถังเดียวกัน ซึ่งไมโครคอนโทรลเลอร์สามารถนำมาประยุกต์ใช้งานได้หลากหลาย โดยผ่านการออกแบบวงจรให้เหมาะกับงานต่าง ๆ และยังสามารถโปรแกรมคำสั่งเพื่อควบคุมขาอินพุต/เอาต์พุต เพื่อควบคุมมอเตอร์ใน



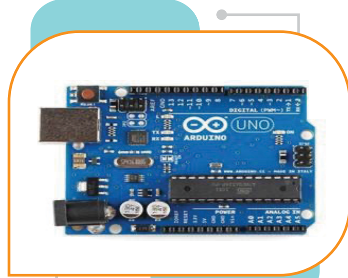
รูปที่ 1.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

หุ่นยนต์ประเภทต่าง ๆ กล่าวได้ว่าไมโครคอนโทรลเลอร์สามารถประยุกต์ใช้งานด้านหุ่นยนต์ ระบบอัตโนมัติ และงานในลักษณะต่าง ๆ



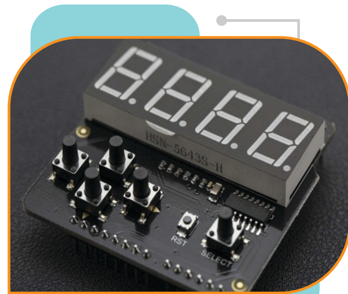
2 บอร์ดอาดุยโน (Arduino)

บอร์ดอาดุยโน (Arduino) รูปที่ 1.2 เป็นไมโครคอนโทรลเลอร์บอร์ดแบบสำเร็จรูปในยุคปัจจุบัน ซึ่งถูกสร้างมาจากคอนโทรลเลอร์ตระกูล AVR โดยข้อดีของบอร์ดไมโครคอนโทรลเลอร์ประเภทนี้ คือ เรื่องของ Open Source ที่สามารถนำไปพัฒนาต่อเป็นอุปกรณ์ต่าง ๆ ได้ และความสามารถในการเพิ่ม Boot Loader เข้าไปที่ตัวไมโครคอนโทรลเลอร์ จึงทำให้การ Upload Code สามารถทำได้ง่ายขึ้น และยังมีการพัฒนาซอฟต์แวร์ ที่ใช้ในการควบคุมตัวบอร์ดของอาดุยโน เป็นภาษา C++ ที่โปรแกรมเมอร์มีความคุ้นเคยในการใช้งาน อีกทั้งบอร์ดยังสามารถนำโมดูลอื่น ๆ เพื่อเพิ่มความสามารถ และใช้งานได้ง่ายขึ้น เรียกว่าเป็นบอร์ดเสริม (Shield) รูปที่ 1.3 และรูปที่ 1.4

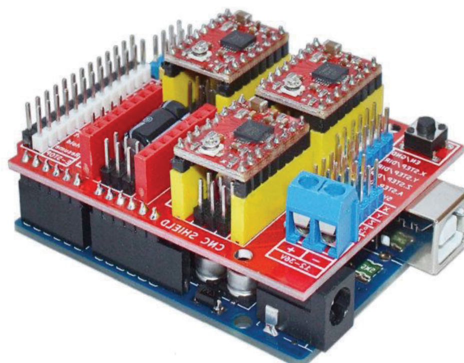


รูปที่ 1.2 บอร์ดอาดุยโน

Model: Arduino UNO R3



รูปที่ 1.3 Keypad Shield For Arduino



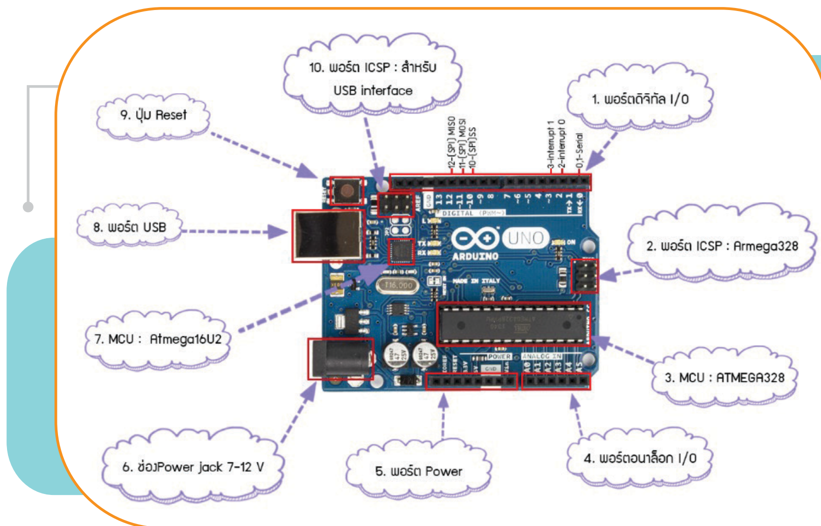
รูปที่ 1.4 CNC Shield V3 for Arduino Uno

2.1 จุดเด่นที่ทำให้บอร์ดอาดุยโน (Arduino)

1. ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
2. มีกลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
3. ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
4. สามารถพัฒนาโปรแกรมบนระบบปฏิบัติการใดก็ได้

2.2 Layout & Pin out Arduino Board (Model: Arduino UNO R3)

ตำแหน่งขา และการจัดวางอุปกรณ์ ของอุปกรณ์บนบอร์ดอาดุยโน (Arduino UNO R3) ดังรูปที่ 1.5



รูปที่ 1.5 *ส่วนประกอบของบอร์ด Arduino UNO R3*

จากรูปที่ 1.5 แสดงถึงส่วนประกอบของบอร์ดอาดุยโน ดังนี้

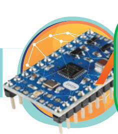
1. I/O Port: ดิจิทัลอินพุต/เอาต์พุต ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บางขาจะทำหน้าที่ อื่น ๆ เพิ่มเติมด้วย เช่น ขา 0,1 เป็นขา Tx, Rx Serial, ขา 3, 5, 6, 9, 10 และ 11 เป็นขา PWM, ขา 10, 11, 12 เป็นขา SPI (SS, MOSI, MISO)
2. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
3. MCU: Atmega328 เป็นไมโครคอนโทรลเลอร์ที่ใช้บนบอร์ดอาดุยโน
4. I/O Port: พอร์ตอนาล็อกอินพุต/เอาต์พุต ตั้งแต่ขา A0 ถึง A5 โดย A4 และ A5 รองรับการเชื่อมต่อสื่อสารแบบ I2C SDA กับ SCL ตามลำดับ
5. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟ +3.3 โวลต์ จำนวน 1 ขา, +5 โวลต์ จำนวน 1 ขา, GND จำนวน 2 ขา, Vin จำนวน 1 ขา
6. Power Jack: รองรับไฟฟ้ากระแสตรงจากภายนอกแรงดันอยู่ระหว่าง 7 ถึง 12V

7 MCU: ของ Atmega16U2 เป็นหน่วยไมโครคอนโทรลเลอร์ที่ทำหน้าที่แปลง USB to Serial โดยจะติดต่อกับคอมพิวเตอร์ผ่าน Atmega16U2

8 USB Port: ใช้สำหรับต่อกับคอมพิวเตอร์ เพื่ออัพโหลดโปรแกรมเข้าไมโครคอนโทรลเลอร์ (MCU) และจ่ายไฟจากพอร์ตคอมพิวเตอร์ให้กับบอร์ด

9 Reset Button: เป็นปุ่มรีเซ็ต (Reset) ใช้กดเมื่อต้องการให้ไมโครคอนโทรลเลอร์เริ่มต้นการทำงานใหม่

10 ICSP Port: เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2



3 บอร์ดอาดุยโนรุ่นต่าง ๆ ที่ได้รับความนิยม

3.1 Arduino UNO R3

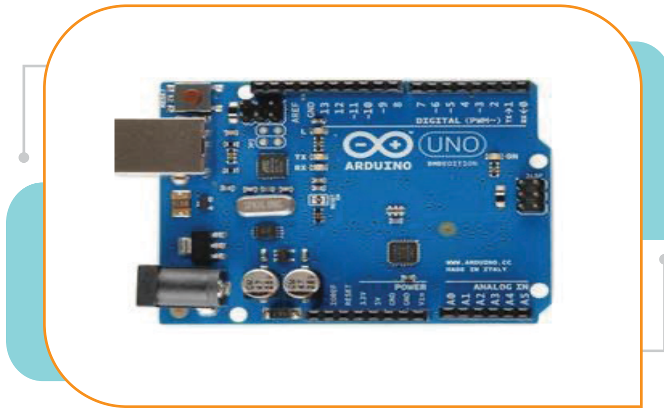
เป็นบอร์ดอาดุยโน (Arduino) ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง และมีการพัฒนาไลบรารี (Library) ต่าง ๆขึ้นมารองรับ (Support) และอ้างอิงกับบอร์ดอาดุยโน ยูโน อาร์3 (Arduino UNO R3) เป็นหลัก ซึ่งมีข้อดีอีกอย่างคือ กรณีที่ ไมโครคอนโทรลเลอร์เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองเนื่องจากเป็นแบบชิพ ถอดได้เปลี่ยนได้ง่าย ดังรูปที่ 1.6



รูปที่ 1.6 บอร์ดอาดุยโน (Arduino UNO R3)

3.2 Arduino Uno SMD

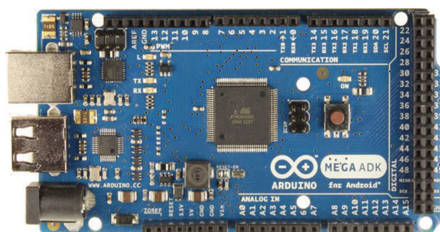
เป็นบอร์ดที่มีคุณสมบัติและการทำงานเหมือนกับบอร์ดอาดุยโน Arduino UNO R3 ทุกประการ แต่จะแตกต่างกับที่ตัว Package ของไมโครคอนโทรลเลอร์ ซึ่งบอร์ดนี้จะมีไมโครคอนโทรลเลอร์ ที่เป็นตัว Package แบบติดตั้งติดกับบอร์ดหลัก SMD (Arduino UNO R3) มีไมโครคอนโทรลเลอร์ที่เป็น Package DIP) ดังรูปที่ 1.7



รูปที่ 1.7 บอร์ดอาดุยโน (Arduino Uno SMD)

3.3 Arduino Mega 2560 R3

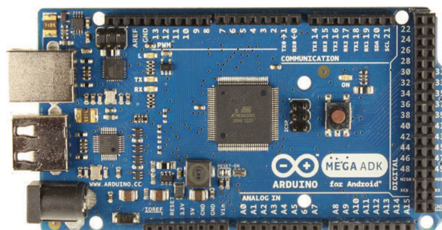
เป็นบอร์ดอาดุยโน (Arduino) ที่ออกแบบมาสำหรับงานที่ต้องใช้อินพุต/เอาต์พุต (I/O) ที่มากกว่า Arduino UNO R3 เช่น งานที่ต้องการรับสัญญาณจากเซนเซอร์หรือควบคุมเซอร์โวมอเตอร์หลาย ๆ ตัว ทำให้อินพุต/เอาต์พุตของบอร์ดอาดุยโน ยูโน อาร์3 (Arduino UNO R3) ไม่สามารถเพียงพอ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีหน่วยความจำแบบ Flash มากกว่าบอร์ดอาดุยโน ยูโน อาร์3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของไมโครคอนโทรลเลอร์ที่เท่ากัน ดังรูปที่ 1.8



รูปที่ 1.8 *บอร์ดอาดุยโน (Arduino Mega 2560 R3)*

3.4 Arduino Mega ADK

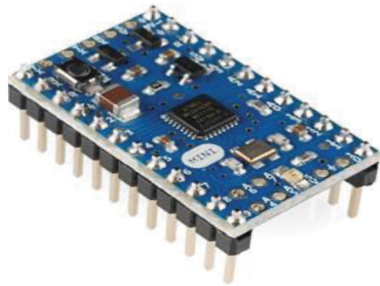
เป็นบอร์ดที่ออกแบบมาให้บอร์ดอาดุยโน Mega 2560 R3 สามารถติดต่อกับอุปกรณ์ Android Device ผ่านพอร์ต USB Host ของบอร์ดได้ ดังรูปที่ 1.9



รูปที่ 1.9 *บอร์ดอาดุยโน (Arduino Mega ADK)*

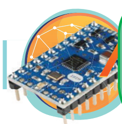
3.5 Arduino Mini 05

เป็นบอร์ดอาดุยโน (Arduino) ขนาดเล็กที่ใช้ไมโครคอนโทรลเลอร์เบอร์ AT-mega328 เบอร์เดียวกับบอร์ด Arduino UNO R3 ดังรูปที่ 1.10



รูปที่ 1.10 *บอร์ดอาดุยโน (Arduino Mini 05)*

ข้อแตกต่าง: บอร์ดอาดุยโน Arduino Mini 05 จะไม่มีพอร์ต USB มาให้ ผู้ใช้งานต้องต่อกับบอร์ด USB to Serial Converter เพิ่ม เมื่อต้องการโปรแกรมบอร์ด




4 การเขียนโปรแกรมภาษาซี สำหรับไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์นอกจาก ฮาร์ดแวร์ (Hardware) นั้น ยังมีส่วนประกอบสำคัญอีกส่วน คือชุดคำสั่งหรือโปรแกรม สำหรับสั่งงานให้ไมโครคอนโทรลเลอร์ทำงานตามที่ต้องการที่เรียกว่า ซอฟต์แวร์ ในการศึกษาจำเป็นต้องเรียนรู้หลักการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

4.1 ขั้นตอนในการพัฒนาโปรแกรม

โปรแกรมคอมพิวเตอร์เป็นการพัฒนาการสร้างกระบวนการควบคุม ซึ่งไม่สามารถเริ่มต้นจากการเขียนคำสั่งด้วยภาษาคอมพิวเตอร์ได้ทันทีสำหรับผู้เรียนรู้ใหม่ไม่มีประสบการณ์การเขียนโปรแกรมควบคุม จำเป็นต้องมีกระบวนการวิเคราะห์ วางแผน และปฏิบัติตามกระบวนการทำงาน แบ่งออกเป็น 5 ขั้นตอน คือ

1. ขั้นตอนการวิเคราะห์ปัญหา โดยจะเริ่มจากการวิเคราะห์ผลลัพธ์ที่ต้องการ (Output) แล้วย้อนกลับไปยังพิจารณาข้อมูลที่นำเข้าสู่ระบบ (Input) ตลอดจนข้อมูลอื่น ๆ ที่เกี่ยวข้องในการที่จะนำไปใช้ในการประมวลผล

2. ขั้นตอนการออกแบบวิธีการแก้ปัญหา เมื่อทราบผลลัพธ์ที่ต้องการและข้อมูลที่นำเข้าสู่ระบบแล้วนั้น ขั้นตอนถัดไปคือ กำหนดการวางแผนในการแก้ปัญหา โดยใช้วิธีเขียนลำดับขั้นตอนการแก้ปัญหาที่เรียกว่า อัลกอริทึม (Algorithm) และใช้เครื่องมือสำหรับช่วยในการเขียนอัลกอริทึมเช่น การเขียน รหัสจำลอง (Pseudo Code) การเขียนผังงาน (Flowchart) เป็นต้น

3. ขั้นตอนการเขียนโปรแกรม เลือกภาษาคอมพิวเตอร์ที่เหมาะสม โดยพิจารณาจากความสามารถของผู้เขียนโปรแกรม และประสิทธิภาพของภาษาคอมพิวเตอร์นั้น ๆ ให้เหมาะสมกับระบบงานที่ต้องการ แล้วเขียนชุดคำสั่งเป็นภาษาคอมพิวเตอร์ตามอัลกอริทึมที่ได้ออกแบบไว้

4. ขั้นตอนการทดสอบและแก้ไขโปรแกรม ภายหลังจากเขียนโปรแกรมเสร็จสิ้น ต้องทำการทดสอบโปรแกรม เพื่อหาข้อผิดพลาด (Error) ซึ่งข้อผิดพลาดที่พบในขั้นตอนการทดสอบโปรแกรมนั้น จะต้องนำมาปรับปรุงแก้ไขโปรแกรมเพื่อให้สามารถทำงานได้

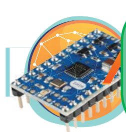
5. ขั้นตอนการจัดทำเอกสารประกอบ อธิบายกระบวนการทำงานของโปรแกรม เมื่อโปรแกรมผ่านการทดสอบแล้ว จะต้องจัดทำเอกสารประกอบ เพื่อความเข้าใจของผู้นำไปใช้งาน ซึ่งจำเป็นต้องมีรายละเอียดของวิธีการใช้งานโปรแกรม วิธีการติดตั้งโปรแกรม ตลอดจนขั้นตอนในการพัฒนาโปรแกรม รวมถึงอัลกอริทึมและโปรแกรมต้นฉบับ (Source code) เพื่อประโยชน์ในกรณีที่ต้องการแก้ไขหรือปรับปรุงโปรแกรมภายหลัง

ขั้นตอนวิธีหรือ Algorithm (ภาษาไทย : อัลกอริทึม) หมายถึงกระบวนการแก้ปัญหาที่สามารถเข้าใจได้มีลำดับ หรือวิธีการในการแก้ไขปัญหาใดปัญหาหนึ่งอย่างเป็นขั้นเป็นตอน และชัดเจน เพื่อให้ได้ผลลัพธ์ที่ต้องการ ซึ่งแตกต่างจากการแก้ปัญหาแบบสามัญสำนึกหรือฮิวริสติก (Heuristic) ซึ่งโดยทั่วไปขั้นตอนวิธีจะประกอบด้วย วิธีการเป็นขั้นตอน และมีการทำงานแบบวนซ้ำ หรือ เวียนเกิดโดยใช้ตรรกะและ/หรือ ในการเปรียบเทียบในขั้นตอนต่าง ๆ จนกระทั่งเสร็จสิ้นการทำงาน ตามผลลัพธ์ที่ต้องการ ในการทำงานอย่างเดียวกันอาจจะเลือกขั้นตอนวิธีการแก้ปัญหาที่แตกต่างกันเพื่อให้ได้ผลลัพธ์ที่ต้องการ ซึ่งสามารถเขียนโปรแกรมให้เหมือนหรือไม่ก็ได้ โดยจะมีความแตกต่างที่จำนวนและชุดคำสั่งที่ใช้ต่างกัน ซึ่งส่งผลให้เวลาและขนาดหน่วยความจำ ที่ต้องการแตกต่างกัน หรือสามารถเรียกได้อีกอย่างว่า มีความ

ผังงาน (Flowchart) คือเป็นขั้นตอนกระบวนการที่อาศัยรูปภาพ (Image) หรือสัญลักษณ์(Symbol) ที่ใช้เขียนแทนขั้นตอน คำอธิบาย ข้อความ หรือคำพูดที่ใช้ในอัลกอริทึม (Algorithm) เพราะการนำเสนอขั้นตอนของงานให้เข้าใจตรงกันระหว่างผู้ที่เกี่ยวข้อง ด้วยคำพูด หรือข้อความทำได้ยากกว่า ซึ่งมีการนำรูปภาพหรือสัญลักษณ์ในข้างต้น เข้ามาประกอบการอธิบาย และแสดงลำดับขั้นตอนการเขียนโปรแกรม โดยผังงานในการเขียนโปรแกรมเป็นรูปทรงเลขาคณิต ที่บรรจุรายละเอียดกระบวนการประมวลผล โดยมีรูปทรงในการใช้งานหลัก ๆ (เฉพาะงาน

รูปทรง	ชนิดการประมวลผล	ตัวอย่างเส้นทาง	
	จุดเริ่มต้นหรือจุดสิ้นสุดของโปรแกรม		
	ประกาศตัวแปร		
	การตัดสินใจ		
	ชุดกระบวนการที่เตรียมไว้ (ฟังก์ชันย่อย)		
	กระบวนการประมวลผล		

รูปทรง	ชนิดการประมวลผล	ตัวอย่างเส้นทาง	
	การแสดงผล		
	จุดเชื่อมต่อภายในหน้าเดียวกัน		
	จุดเชื่อมต่อระหว่างหน้า		

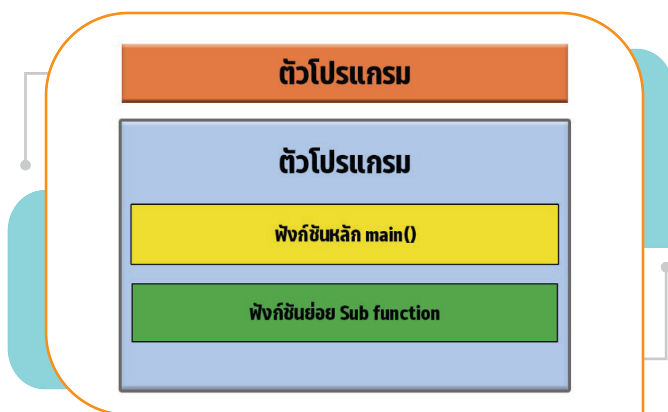



ภาษาซีสำหรับไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ ไม่ว่าจะเป็นตระกูลใดก็ตามจะทำงานก็ต่อเมื่อมีชุดคำสั่งที่สั่งให้ทำงานตามที่ต้องการที่เรียกว่า โปรแกรม โดยคำสั่งหรือโปรแกรมที่ไมโครคอนโทรลเลอร์เข้าใจและสามารถทำงานได้จะอยู่ในรูปของลอจิก 0 และ 1 หากนำลอจิกมาจับกลุ่มก็เป็นเลขฐาน 16 ที่เรียกว่า ภาษาเครื่อง ซึ่งภาษาเครื่องเป็นภาษาที่มนุษย์ไม่สามารถเข้าใจได้เนื่องจากเป็นเลขฐาน 16 ทั้งหมด ดังนั้นในการเขียนโปรแกรมจึงจำเป็นต้องใช้ภาษาที่มนุษย์สามารถเข้าใจได้ โดยภาษาที่มนุษย์เข้าใจได้และใกล้เคียงกับภาษาเครื่องมากที่สุด คือภาษาแอสเซมบลี แต่เนื่องจากการพัฒนางานโดยใช้ภาษาแอสเซมบลีเป็นไปได้ยากและซับซ้อน ดังนั้นเพื่อให้ง่ายและรวดเร็วต่อการพัฒนาโปรแกรมใช้งานไมโครคอนโทรลเลอร์ภาษาที่เหมาะสมคือภาษาซี

5.1 โครงสร้างของภาษาซี

ภาษาซีเป็นภาษาที่ได้รับความนิยมสูงเป็นภาษาโครงสร้างง่ายต่อการทำความเข้าใจง่ายต่อการนำไปพัฒนา สามารถเขียนโปรแกรมแยกเป็นส่วน ๆ โดยแต่ละส่วนสามารถเรียกใช้งาน จากส่วนอื่นของโปรแกรม ทำให้สามารถแบ่งงานให้หลายคนช่วยกันพัฒนาได้การเขียนโปรแกรมเป็นส่วน ๆ เรียกว่า ฟังก์ชัน โดยโครงสร้างของภาษาซีมีส่วนประกอบ 2 ส่วนคือ ส่วนหัวโปรแกรมและส่วนตัวโปรแกรม ดังรูปที่ 1.11 เริ่มต้นจาก ส่วนตัวโปรแกรม จะมีฟังก์ชันหลัก ชื่อว่า main() เพื่อเป็นส่วนหลักและเป็นส่วนแรกของการเริ่มต้นการทำงาน และอาจมีฟังก์ชันอื่นที่ผู้พัฒนาเขียนขึ้นเพื่อใช้งานเป็นส่วน ๆ เรียกว่า ฟังก์ชันรอง หรือฟังก์ชันย่อย



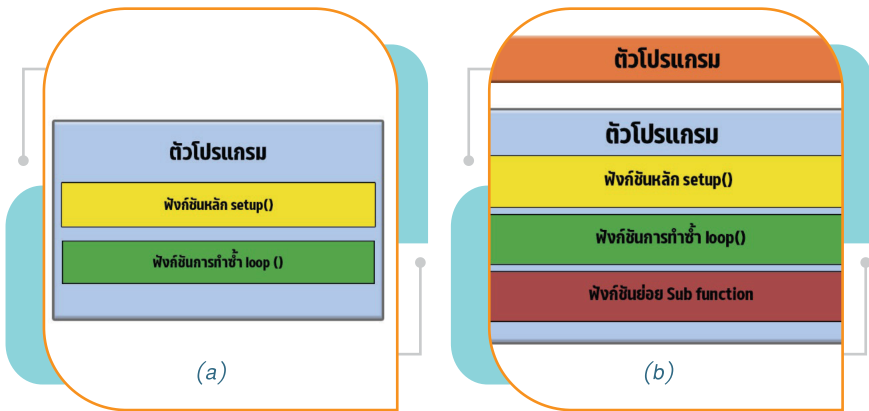
รูปที่ 1.11 CNC Shield V3 for Arduino Uno

5.2 โครงสร้างของภาษาซีสำหรับอาดุยโน (Arduino)

โครงสร้างภาษาซีสำหรับอาดุยโน ถูกจัดวางใหม่ให้ง่ายต่อการพัฒนาโปรแกรมของผู้ใช้งานเบื้องต้น ซึ่งผู้ออกแบบได้จัดวางให้ผู้พัฒนาได้ใช้งานได้ง่าย โดยโครงสร้างหลัก ๆ จะมีเพียง 2 ส่วนเท่านั้น คือ

1 setup เป็นส่วนที่เก็บฟังก์ชัน ที่ทำงานเพียงครั้งเดียว

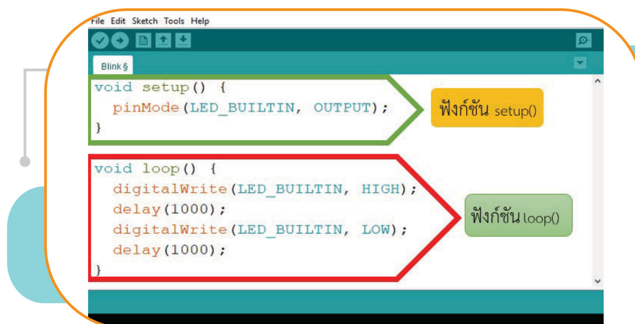
2 loop เป็นส่วนที่เก็บฟังก์ชัน ที่เมื่อทำงานครบแล้วจะวนกลับมาทำงานซ้ำตั้งแต่ต้น ดังรูปที่ 1.12(a) แต่ถ้าต้องการเขียนโปรแกรมขั้นสูงก็สามารถเขียนในส่วนหัวโปรแกรมและส่วนของฟังก์ชันย่อยที่เขียนขึ้นใช้งานเอง ดังรูปที่ 1.12(b) เพื่อความสะดวก และความเข้าใจ ซึ่งเป็นวิธีการเดียวกับภาษาซีมาตรฐาน



รูปที่ 1.12 โครงสร้างการเขียนโปรแกรมด้วยภาษาซีสำหรับอาดุยโน

จากรูปที่ 1.12 โครงสร้างการเขียนโปรแกรมด้วยภาษาซีสำหรับอาดุยโน จะสามารถเขียนโปรแกรมในโปรแกรม Arduino IDE ที่ใช้ในการพัฒนาโปรแกรมบนบอร์ดตระกูลอาดุยโน โดยสามารถเขียนโปรแกรมตามโครงสร้าง ได้ดังตัวอย่าง ดังรูปที่ 1.13

ตัวอย่างโปรแกรมที่เขียนด้วยโครงสร้างแบบพื้นฐาน

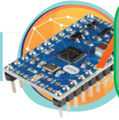


รูปที่ 1.13 ตัวอย่างโปรแกรมที่เขียนด้วยโครงสร้างแบบพื้นฐาน

ตัวอย่างโปรแกรมที่เขียนด้วยโครงสร้างแบบพื้นฐาน



รูปที่ 1.14 ตัวอย่างโปรแกรมที่เขียนด้วยโครงสร้างแบบขั้นสูง



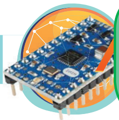
6 ตัวแปรในภาษาซี

ตัวแปรคือการประกาศที่จะใช้ในโปรแกรม ซึ่งจะส่งผลต่อการจองพื้นที่ในหน่วยความจำ เพื่อนำไปใช้งานในฟังก์ชัน โดยการใช้ชื่อตัวแปรแทนการกำหนดเป็นค่าแอดเดรสของหน่วยความจำรูปแบบของการประกาศตัวแปรเป็นดังนี้

```
ชนิดของตัวแปร  ชื่อตัวแปร;
```

หรือ

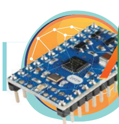
```
ชนิดของตัวแปร  ชื่อตัวแปรที่1,ชื่อตัวแปรที่2,... ;
```



7 หลักการตั้งชื่อตัวแปรในภาษาซี

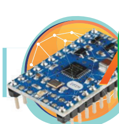
หลักการตั้งชื่อตัวแปรมีข้อกำหนดหลัก ๆ อยู่ 4 ประการ คือ

1. ชื่อที่ตั้งต้องไม่ซ้ำ กับคำสงวนของภาษาซี (คำที่ภาษาซีมีใช้งานอยู่แล้ว)
2. การใช้ตัวอักษรใหญ่กับตัวอักษรเล็กถือว่า เป็นคนละตัวกัน
3. ตัวแรกของชื่อตัวแปรต้องเป็นตัวอักษรเท่านั้น ตัวถัดไปจะเป็นตัวเลขก็ได้
4. ชื่อตัวแปรห้ามเว้นวรรค ยกเว้นใส่ขีดล่าง _



8 ชนิดของตัวแปรในภาษาซีสำหรับอาดุยโน

ชนิด	ขนาด บิต	ขอบเขต
char (ตัวอักขระ)	8	-128 ถึง +127
unsigned char (ตัวอักขระไม่คิดเครื่องหมาย)	8	0 ถึง 255
int (จำนวนเต็ม)	16	-32768 ถึง +32767
unsigned int (จำนวนเต็มไม่คิดเครื่องหมาย)	16	0 ถึง 65535
long (จำนวนเต็มแบบยาว)	32	-2147483648 ถึง +2147483649
unsigned long (จำนวนเต็มแบบยาว ไม่คิดเครื่องหมาย)	32	0 ถึง 4294967295
fl oat (จำนวนจริง)	32	3.4E-38 ถึง 3.4E+38 หรือ ทศนิยม 6
double (จำนวนจริง)	64	+3.4E\38 (~7 digits)



9 ตัวดำเนินการในภาษาซี

ตัวดำเนินการในภาษาซีแบ่งตามลักษณะการกระทำได้ 5 กลุ่ม คือ

1. ตัวกระทำทางคณิตศาสตร์
2. ตัวกระทำทางลอจิกระดับบิต
3. ตัวกระทำบูลีน
4. ตัวกระทำเปรียบเทียบ
5. ตัวกระทำประสม

โดยในแต่ละลักษณะมีรายละเอียดดังต่อไปนี้

ตารางที่ 1.3 ตัวกระทำทางคณิตศาสตร์

เครื่องหมาย	การกระทำ	ตัวอย่าง	คำอธิบาย
+	บวก	x=y+z;	x เท่ากับ ค่าในตัวแปร y บวกกับค่าในตัวแปร z
-	ลบ	x=y-z;	x เท่ากับ ค่าในตัวแปร y ลบด้วยค่าในตัวแปร z
*	คูณ	x=y*z;	x เท่ากับ ค่าในตัวแปร y คูณด้วยค่าในตัวแปร z
/	หาร	x=y/z;	x เท่ากับ ค่าในตัวแปร y หารด้วยค่าในตัวแปร z
%	หารเอาเศษ	x=y%z;	x เท่ากับ เศษของการหารระหว่างตัวแปร y กับตัวแปร z

ตารางที่ 1.4 ตัวกระทำทางลอจิก

เครื่องหมาย	การกระทำ	ตัวอย่าง	คำอธิบาย
&	แอนด์	x=y&z;	x เท่ากับค่าในตัวแปร y แอนด์กับค่าในตัวแปร z
|	ออร์	x=y|z;	x เท่ากับค่าในตัวแปร y ออร์กับค่าในตัวแปร z
^	เอ็กซ์คลูซิฟออร์	x=y^z;	x เท่ากับค่าในตัวแปร y เอ็กซ์คลูซิฟออร์กับตัวแปร z
~	วันคอมพลีเมนต์	x=~y;	x เท่ากับค่าตรงข้ามของค่าในตัวแปร y
<<	เลื่อนไปทางซ้าย	x=x<<1;	เลื่อนข้อมูลใน x ไปทางซ้ายไป 1 บิต
>>	เลื่อนไปทางขวา	x=x>>2	เลื่อนข้อมูลใน x ไปทางขวาไป 2 บิต

ตารางที่ 1.5 ตัวกระทำบูลีน

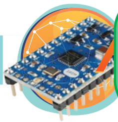
เครื่องหมาย	การกระทำ	คำอธิบาย
&&	แอนด์	เชื่อมเงื่อนไข 2 เงื่อนไขด้วยคำว่า "และ"
||	ออร์	เชื่อมเงื่อนไข 2 เงื่อนไขด้วยคำว่า "หรือ"
!	อินเวิร์ส	คือ Not หรือ ไม่ ใช้ยกเลิกสถานะตรรกะ เช่น ถ้าเงื่อนไขเป็นจริง แล้วมีเครื่องหมาย ! ข้างหน้า เงื่อนไขนั้นจะกลายเป็นเท็จ

ตารางที่ 1.6 ตัวกระทำการเปรียบเทียบ

เครื่องหมาย	การกระทำ	ตัวอย่าง	คำอธิบาย
>	มากกว่า	x>5	x มากกว่า 5
<	น้อยกว่า	x<5	x น้อยกว่า 5
>=	มากกว่าหรือเท่ากับ	x>=5	x มากกว่าหรือเท่ากับ 5
<=	น้อยกว่าหรือเท่ากับ	x<=5	x น้อยกว่าหรือเท่ากับ 5
==	เท่ากับ	x==5	x เท่ากับ 5
!=	ไม่เท่ากับ	x!=5	x ไม่เท่ากับ 5

ตารางที่ 1.7 ตัวกระทำประสม

เครื่องหมาย	การกระทำ	ตัวอย่าง	คำอธิบาย
++	เพิ่มค่า 1 ค่า	x++;	เพิ่มค่า x ขึ้น 1 ค่า
--	ลดค่า 1 ค่า	x--;	ลดค่า x ลง 1 ค่า
+=	บวก	x+=2;	x ใหม่เท่ากับ x เดิมบวกกับ 2
-=	ลบ	x-=2;	x ใหม่เท่ากับ x เดิมลบด้วย 2
*=	คูณ	x*=2;	x ใหม่เท่ากับ x เดิมคูณด้วย 2
/=	หาร	x/=2;	x ใหม่เท่ากับ x เดิมหารด้วย 2
%=	หารเอาเศษ	x%=2;	x ใหม่เท่ากับ x เดิมหารด้วย 2 แล้วเอาเศษ
&=	แอนด์	x&=2;	x ใหม่เท่ากับ x เดิมแอนด์ด้วย 2
|=	ออร์	x|=2;	x ใหม่เท่ากับ x เดิมออร์ด้วย 2



10 ฟังก์ชันการดำเนินการแบบทางเลือก

ในการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ทำงานตามที่ต้องการ นอกจากฟังก์ชัน ที่สั่งให้ทำงานเป็นลำดับแล้วนั้น การเขียนโปรแกรมจำเป็นต้องใช้ฟังก์ชันที่มีการทำ งานแบบเลือกเส้นทางการทำงาน โดยการทำตามเงื่อนไข หรือการให้ทำซ้ำแบบมีเงื่อนไข โดยฟังก์ชันที่มีการทำงานแบบทางเลือกในภาษาซีมีด้วยกัน 4 ฟังก์ชัน คือ

1. ฟังก์ชัน if
2. ฟังก์ชัน if-else
3. ฟังก์ชัน if else-if
4. ฟังก์ชัน switch...case

10.1 ฟังก์ชัน if (ทางเลือกเดียว)

ฟังก์ชัน if เป็นฟังก์ชันที่มีการตรวจสอบเงื่อนไข ซึ่งถ้าเงื่อนไขเป็นจริงจะทำงานตามชุดฟังก์ชันที่กำหนดไว้รูปแบบ ดังนี้

ตารางที่ 1.8 if (เงื่อนไขที่ตรวจสอบ)

ผังงาน	โค้ดโปรแกรม
เงื่อนไข → ฟังก์ชันทำงาน เมื่อเงื่อนไขถูกต้อง	if (conditional) {
	// put your code here
	// if conditional true
	}
	Example1:
	if (value<25) {
	digitalWrite(10, HIGH);
	}
	//หากชุดฟังก์ชัน ที่ให้ทำงานเมื่อเงื่อนไขเป็นจริงมีเพียงฟังก์ชัน เดียวไม่ต้องใส่วงเล็บปีกกา {...} ก็ได้
	Example2:
	if (value<25) digitalWrite(10, HIGH);

10.2 ฟังก์ชัน if-else (สองทางเลือก)

การตรวจสอบเงื่อนไขที่มีชุดฟังก์ชัน ให้ทำงานเมื่อเงื่อนไขถูก และมีชุดฟังก์ชันให้ทำงานเมื่อเงื่อนไขผิด จะใช้ฟังก์ชัน if-else มาใช้งาน ซึ่งมีรูปแบบดังนี้

ตารางที่ 1.9 if (เงื่อนไขที่ตรวจสอบ) else

ผังงาน	โค้ดโปรแกรม
เงื่อนไข	if (conditional)
ฟังก์ชันทำงานเมื่อเงื่อนไขถูกต้อง	{
ฟังก์ชันทำงานเมื่อเงื่อนไขผิด	// put main code here
	// if conditional true
	}
	else
	{
	//put main code here
	//if conditional false
	}

10.3 ฟังก์ชัน if else-if (หลายทางเลือก)

ฟังก์ชันที่มีการตรวจสอบเงื่อนไขหลายเงื่อนไข และมีชุดฟังก์ชัน ที่เตรียมให้ทำงานในแต่ละเงื่อนไขหากเงื่อนไขนั้น ๆ ถูกต้อง

ตารางที่ 1.10 if (เงื่อนไขที่ตรวจสอบ) else if (เงื่อนไขที่ตรวจสอบ)

ผังงาน	โค้ดโปรแกรม
เงื่อนไขที่ 1 เงื่อนไขที่ 2 ฟังก์ชันทำงาน เมื่อเงื่อนไขถูกต้อง 1 ฟังก์ชันทำงาน เมื่อเงื่อนไขถูกต้อง 2	if (conditional)
	{
	// put main code here
	// if conditional true
	}
	Else if (conditional)
	{
	//put main code here
	//if conditional false
	}
	else
	{
	//put main code here
	//if conditional false
	}

10.4 ฟังก์ชัน if else-if (หลายทางเลือก)

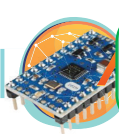
ฟังก์ชัน switch...case เป็นฟังก์ชัน หลายทางเลือกอีกฟังก์ชันหนึ่ง ที่มีการทำงานคล้ายกับฟังก์ชัน if else-if ต่างตรงที่การตรวจสอบเงื่อนไข จะมีวิธีการตรวจสอบการเท่ากันของเงื่อนไขตัวแปรที่ใช้ตรวจสอบเท่านั้น โดยฟังก์ชัน switch...case มีคำสั่งที่สำคัญอยู่ 3 คำสั่ง คือ

1. switch เป็นคำสั่งที่ใช้ตรวจสอบเงื่อนไข เพื่อใช้ในการเลือก case ในการทำงาน
2. case เป็นชุดทางเลือก หากตรวจสอบเงื่อนไข switch แล้วเงื่อนไขของ case นั้น เป็นจริง โปรแกรมจะเข้าไปทำงานต่อใน case นั้น ๆ
3. break เป็นคำสั่งที่จะอยู่ใน case ทุก case คือเมื่อทำงานตามคำสั่งตามเงื่อนไขแล้วนั้น break จะทำให้โปรแกรมหยุดทำงาน แต่หากไม่มีคำสั่ง break ใน case จะทำให้โปรแกรมจะทำงานต่อไปเรื่อย ๆ ไม่มีที่สิ้นสุด

ตารางที่ 1.11 switch...case (หลายทางเลือก)

ผังงาน	โค้ดโปรแกรม
ตัวแปรที่ใช้ในการตรวจสอบ เงื่อนไขที่ 1 เงื่อนไขที่ 2 ตรงกับค่าที่กำหนด 1 ตรงกับค่าที่กำหนด 2 ฟังก์ชันที่ไม่ตรงกับค่าที่กำหนดเลย Break(); Break(); Break();	switch (variable)
	{
	case 1:
	// put code here for case 1
	break;
	case 2:
	// put code here for case 2
	break;

ผังงาน	โค้ดโปรแกรม
	`default:`
	`// put code here for default`
	`break;`
	`}`



11 ฟังก์ชันการดำเนินการแบบวนซ้ำ

การเขียนโปรแกรมสั่งงานไมโครคอนโทรลเลอร์ ในบางโปรแกรมที่พัฒนาขึ้น จะมีการทำงานแบบวนซ้ำหรือวนรอบ เพื่อที่จะทำงานในชุดคำสั่งเดิม ๆ โดยลักษณะการทำงานมีทั้งแบบมีเงื่อนไขหรือไม่มีเงื่อนไขก็ได้ โดยในภาษาซีจะมีฟังก์ชันสั่งงานให้ไมโครคอนโทรลเลอร์ทำงานซ้ำหลัก ๆ อยู่ 3 รูปแบบ ดังนี้

1. ฟังก์ชัน for
2. ฟังก์ชัน while
3. ฟังก์ชัน do-while

11.1 ฟังก์ชัน for

ฟังก์ชัน for เป็นฟังก์ชันที่ใช้ในกรณีที่ทราบจำนวนรอบที่จะทำงานซ้ำแน่นอน เช่น 5 รอบ, 10 รอบ เป็นต้น

ตารางที่ 1.12 ตัวอย่างการใช้งานฟังก์ชัน for

ผังงาน	โค้ดโปรแกรม
กำหนดค่าตัวแปรรอบเริ่มต้น เงื่อนไขที่ ฟังก์ชันที่ต้องการทำซ้ำ เพิ่ม/ลดตัวแปรนับรอบ	for(ค่าเริ่มต้น; เงื่อนไข; เพิ่มหรือลดค่า)
	{
	//ชุดฟังก์ชัน ที่ต้องการทำซ้ำ
	}

ตารางที่ 1.13 ตัวอย่างการเขียนโปรแกรมฟังก์ชัน for

โค้ดโปรแกรม	คำอธิบาย
for(int i=0; i<5; i++)	ประกาศและกำหนดตัวแปรนับรอบเป็นตัวแปร i เป็นตัวแปรชนิดจำนวนเต็ม (integer) โดยกำหนดค่าเริ่มต้น เท่ากับศูนย์ทำวนซ้ำไปเรื่อย ๆ หากค่าตัวแปรยังน้อยกว่า 5 โดยรอบถัดไปให้เพิ่มค่าในตัวแปรนับรอบขึ้น 1 ค่า จนกว่าเงื่อนไขที่กำหนดผิด
{	
digitalWrite(5,HIGH);	
delay(1000);	
digitalWrite(5,LOW);	
delay(1000);	
}	

1.2 ฟังก์ชัน while

ฟังก์ชัน while เป็นฟังก์ชันที่ให้ทำงานวนซ้ำ หรือวนรอบโดยมีการตรวจสอบเงื่อนไขก่อนถ้าหากเงื่อนไขถูกจะทำงานตามชุดฟังก์ชันที่เตรียมไว้ เมื่อทำงานในชุดฟังก์ชันเสร็จ จะมีการวนกลับไปตรวจสอบเงื่อนไขอีกครั้ง ซึ่งตรวจสอบแล้วเงื่อนไขผิด จะออกจากวงรอบการทำซ้ำ

ตารางที่ 1.14 ตัวอย่างการใช้งานฟังก์ชัน while

ผังงาน	โค้ดโปรแกรม
กำหนดค่าตัวแปรรอบเริ่มต้น เงื่อนไขที่ ฟังก์ชันที่ต้องการทำซ้ำ	while(เงื่อนไข)
	{
	//ชุดฟังก์ชัน ที่ต้องการทำซ้ำ
	}

ตารางที่ 1.15 ตัวอย่างการเขียนโปรแกรมฟังก์ชัน while

โค้ดโปรแกรม	คำอธิบาย
`i=0;`	ตรวจสอบก่อนว่าเงื่อนไขเป็นจริงอยู่หรือไม่ หากตัวแปร i ยังน้อยกว่า 5 หากเงื่อนไขถูกให้ทำงานในฟังก์ชันที่ต้องการทำซ้ำ เมื่อทำงานครบให้กลับมาตรวจสอบเงื่อนไขใหม่วนซ้ำไปเรื่อย ๆ จนกว่าเงื่อนไขผิด ในตัวอย่างนี้ คือ มากกว่าหรือเท่ากับ 5
`while(i<5)`	
`{`	
`digitalWrite(5,HIGH);`	
`delay(1000);`	
`digitalWrite(5,LOW);`	
`delay(1000);`	
`i++;}`	

11.3 ฟังก์ชัน do-while

ฟังก์ชัน do-while เป็นฟังก์ชันที่ให้ทำงานวนซ้ำ หรือวนรอบ โดยมีการตรวจสอบเงื่อนไขคล้ายกับฟังก์ชัน while แต่มีต่างตรงที่ฟังก์ชัน do-while จะทำงานตามคำสั่งในลูป do ให้เรียบร้อยก่อน แล้วจึงจะทำการพิจารณาเงื่อนไขในเงื่อนไขของ while หากเงื่อนไขของ while เป็นจริง จะทำให้ลูป do ซ้ำ ๆ ไปเรื่อย ๆ จนกว่าเงื่อนไขของ while เป็นเท็จ

ตารางที่ 1.16 ตัวอย่างการใช้งานฟังก์ชัน do-while

ผังงาน	โค้ดโปรแกรม
กำหนดค่าตัวแปรรอบเริ่มต้น เงื่อนไขที่	do
	{
	//ชุดฟังก์ชัน ที่ต้องการทำซ้ำ
	} while(เงื่อนไข)

ตารางที่ 1.17 ตัวอย่างการเขียนโปรแกรมฟังก์ชัน do-while

โค้ดโปรแกรม	คำอธิบาย
i=0;	ทำงานในฟังก์ชัน ที่เตรียมไว้แล้วตรวจสอบเงื่อนไขตัวแปร i ยังมีค่าน้อยกว่า 5 หรือไม่ หากน้อยกว่าให้วนกลับไปทำซ้ำ ๆ จนกว่า เงื่อนไขจะผิด
do	
{	
digitalWrite(5,HIGH);	
delay(1000);	
digitalWrite(5,LOW);	
delay(1000);	
i++;	
} while(i<5)	

แบบฝึกหัดท้ายบทที่ 1

วิชา : ชุดการเรียนรู้เริ่มต้นการเขียนโปรแกรมและการต่อวงจรสำหรับ Aruidno	
เรื่อง : ความรู้เบื้องต้นเกี่ยวกับอาดุยโน (Arduino) และการเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์	ใช้เวลา 20 นาที

คำชี้แจง 1. แบบทดสอบมีทั้งหมด 20 ข้อ (20 คะแนน)
2. ให้ผู้เรียนเลือกคำตอบที่ถูกที่สุดแล้วกาเครื่องหมายกากบาท (x) ลงในกระดาษคำตอบ

1. ข้อใดไม่ใช่โครงสร้างของไมโครคอนโทรลเลอร์
   ก. หน่วยความจำถาวร ข. หน่วยความจำชั่วคราว
   ค. อินพุต/เอาต์พุต ง. เครือข่ายอินเทอร์เน็ต
2. บอร์ดอาดุยโน (Arduino) ใช้ไมโครคอนโทรลเลอร์ตระกูลอะไร
   ก. ตระกูล MSC-51 ข. ตระกูล AVR
   ค. ตระกูล BASIC STOP ง. ตระกูล PICTURE
3. บอร์ดอาดุยโน ยูโน อาร์3 (Arduino Uno R3) ใช้ไมโครคอนโทรลเลอร์เบอร์อะไร
   ก. Atmega32 ข. Atmega2560
   ค. Atmega32U4 ง. Atmega328
4. บอร์ดอาดุยโน ยูโน อาร์3 (Arduino Uno R3) มีอินพุต/เอาต์พุต ดิจิทัลจำนวนกี่พอร์ต
   ก. 13 พอร์ต ข. 14 พอร์ตค. 15 พอร์ต ง. 16 พอร์ต
5. บอร์ดอาดุยโน ยูโน อาร์3 (Arduino Uno R3) ใช้ไมโครคอนโทรลเลอร์อะไรที่ทำหน้าที่เป็น USB to Serial Interface
   ก. Atmega128 ข. Atmega32U4
   ค. Atmega16U2 ง. Atmega2560

6. ความแตกต่างระหว่าง Arduino Uno R3 กับ Arduino Uno SMD ข้อใดถูกต้อง
   ก. Package ของไมโครคอนโทรลเลอร์
   ข. ขนาดบอร์ด
   ค. จำนวนพอร์ตดิจิทัล Arduino Uno SMD มากกว่า
   ง. จำนวนพอร์ตอนาล็อก Arduino Uno R3 มากกว่า
7. ขั้นตอนในการพัฒนาโปรแกรม ขั้นตอนใดที่ใช้วางแผนลำดับขั้นตอนวิธีการแก้ปัญหา
   ก. ขั้นตอนการวิเคราะห์ปัญหา
   ข. ขั้นตอนการออกแบบวิธีการแก้ปัญหา
   ค. ขั้นตอนการเขียนโปรแกรม
   ง. ขั้นตอนการทดสอบและแก้ไขโปแกรม
8. ขั้นตอนในการพัฒนาโปรแกรม ขั้นตอนใดที่จะให้ผู้นำไปใช้งานมีความเข้าใจในการทำงานของโปรแกรม สามารถใช้งาน และพัฒนาต่อได้
   ก. ขั้นตอนการจัดทำเอกสารประกอบ ข. ขั้นตอนการวิเคราะห์ปัญหา
   ค. ขั้นตอนการเขียนโปรแกรม ง.ขั้นตอนการทดสอบและแก้ไขโปแกรม
9. บอร์ดอาดุยโน ยูโน อาร์3 (Arduino Uno R3) มีอินพุต/เอาต์พุต อนาล็อกจำนวนกี่พอร์ต
   ก. 5 พอร์ต ข. 6 พอร์ต ค. 7 พอร์ต ง. 8 พอร์ต
10. ผังการทำงาน (Flowchart) ดังรูป ◆ คือชนิดการประมวลผลประเภทใด
   ก. ประกาศตัวแปร
   ข. จุดเริ่มต้นหรือจุดสิ้นสุดของโปรแกรม
   ค. การแสดงผล
   ง. การตัดสินใจ

11. ผังการทำงาน(Flowchart)ดังรูป คือชนิดการประมวลผลประเภทใด

ก. ประกาศตัวแปร ข. การแสดงผล

ค. ฟังก์ชันย่อย ง. การตัดสินใจ

12. ข้อใดไม่ใช่หลักการตั้งชื่อตัวแปรในภาษาซี

ก. ชื่อที่ตั้งต้องไม่ซ้ำ กับคำสงวนของภาษาซี (คำที่ภาษาซีมีใช้งานอยู่แล้ว)

ข. การใช้ตัวอักษรใหญ่กับตัวอักษรเล็กถือว่า เป็นคนละตัวกัน

ค. ตัวแรกของชื่อตัวแปรต้องเป็นตัวเลข

ง. ชื่อตัวแปรห้ามเว้นวรรค ยกเว้นใส่ขีดล่าง _

13. การตั้งชื่อตัวแปรในภาษาซีข้อใดผิด

ก. else ข. Else ค. Aimphan ง. ABC123

14. หากต้องการประกาศตัวแปรเป็นตัวแปรชนิดจำนวนเต็ม ขนาด 16 บิต ซึ่งมีขอบเขตเป็นตัวเลขทั้งลบ และบวก จำเป็นต้องประกาศตัวแปรชนิดใด

ก. unsigned char ข. int

ค. unsigned int ง. float

15. ชนิดของตัวแปรประเภทไหนที่สามารถเก็บข้อมูลได้สูงที่สุด

ก. long ข. float ค. double ง. unsigned int

16. x=x<<3; มีความหมายว่าอย่างไร

ก. เลื่อนข้อมูลใน x ไปทางขวาไป 1 บิต จำนวน 3 ครั้ง

ข. เลื่อนข้อมูลใน x ไปทางขวาไป 3 บิต

ค. เลื่อนข้อมูลใน x ไปทางขวาไป 2 บิต

ง. เลื่อนข้อมูลใน x ไปทางซ้ายไป 3 บิต

17. การกำหนดเงื่อนไขแบบตัวกระทำบูลีน การกระทำแบบใด ที่เงื่อนไขต้องเป็นจริงทั้งคู่ ถึงจะทำงาน

ก. != ข. || ค. && ง. >=

18. delay(1000); มีความหมายว่าอย่างไร
   ก. หน่วงเวลา 1 วินาที
   ข. หน่วงเวลา 0.1 วินาที
   ค. หน่วงเวลา 100 วินาที
   ง. หน่วงเวลา 1000 วินาที

19. ฟังก์ชันการดำเนินการแบบวนซ้ำ เป็นฟังก์ชันที่ใช้ในกรณีที่ทราบจำนวนรอบที่จะทำงานซ้ำแน่นอน คือฟังก์ชันใด
   ก. switch-case
   ข. do-while
   ค. while
   ง. for

20. ฟังก์ชัน while กับฟังก์ชัน do-while แตกต่างกันอย่างไร
   ก. ทำงานเหมือนกัน
   ข. ทำงานเมื่อเงื่อนไขเป็นจริงเท่านั้น
   ค. ทั้ง 2 ฟังก์ชันทำงานเพียงรอบเดียว
   ง. ฟังก์ชัน do-while จะทำงานในฟังก์ชันที่เตรียมไว้ 1 รอบ แล้วค่อยตรวจสอบเงื่อนไข